Semi-Supervised Classification Using Tree-Based Self-Organizing Maps

[1] Universidad de Talca, Km. 1 Camino a Los Niches, Curicó, Chile
castudillo@utalca.cl,
http://ing.utalca.cl/~castudillo/
[2] School of Computer Science, Carleton University, Ottawa, Canada, K1S 5B6
oommen@scs.carleton.ca
http://scs.carleton.ca/~oommen/

Abstract. This paper presents a classifier which uses a tree-based Neural Network (NN), and uses both, unlabeled and labeled instances. First, we learn the structure of the data distribution in an unsupervised manner. After convergence, and once labeled data become available, our strategy tags each of the clusters according to the evidence provided by the instances. Unlike other neighborhood-based schemes, our classifier uses only a small set of representatives whose cardinality can be much smaller than that of the input set. Our experiments show that, on average, the accuracy of such classifier is reasonably comparable to those obtained by some of the state-of-the-art classification schemes that only use labeled instances during the training phase. The experiments also show that improved levels of accuracy can be obtained by imposing trees with a larger number of nodes.

Keywords: Hierarchical SOM, Topology-Based Self-Organization, Pattern Recognition, Semi-Supervised Learning.

1 Introduction

The literature includes scores of algorithms which can achieve supervised Pattern Recognition (PR) [5]. Such schemes assume the full specification of the identity of each training instance. In the unsupervised model [10], the class labels of the instances are assumed to be unknown. Rather, the algorithm attempts to infer the distinct group of items, a process which might be time consuming, especially for large datasets. We propose using the Tree-based Topology Oriented SOM (TTOSOM) [1] for classification, attempting to bridge the two paradigms

⋆ The work of this author was done while pursuing his Doctoral studies at the School of Computer Science, Carleton University, Ottawa, Canada : K1S 5B6.

⋆⋆ *Chancellor's Professor* ; *Fellow : IEEE* and *Fellow : IAPR*. This author is also an *Adjunct Professor* with the University of Agder in Grimstad, Norway. The work of this author was partially supported by NSERC, the Natural Sciences and Engineering Research Council of Canada.

D. Wang and M. Reynolds (Eds.): AI 2011, LNAI 7106, pp. 21–30, 2011.

by following the so-called semi-supervised approach, explained by Zhu [15]. Using an unsupervised approach, we will first train a TTOSOM in which the neural tree mimics the properties of the input set. Subsequently, we assign a class label to every single node in the NN[1] by using a voting scheme.

On receiving the testing data, one determines the closest neuron to the testing sample and assigns the sample to the corresponding class.

Once the TTOSOM has been computed, the complexity of the testing is linear, not in cardinality of the training set, but in the size of the neural tree.

Such a nearest-neighbor type testing is similar to the ones used by prototype reduction schemes [13]. However, in our case, the prototypes are not unrelated to each other. Rather, they are constrained by the tree structure.

We cannot expect an accuracy greater than that which a true nearest neighbor classifier yields, because we could be only using a small set (e.g., 25 prototypes) instead of the entire set, which could consist of thousands of points. Further, by starving the classifier of information of the class labels, one can expect the accuracy to be even less. What is astonishing, however, is the fact that our "semi-supervised" TTOSOM-based classifier achieves an accuracy which is only marginally less than state-of-the-art supervised classifiers reported.

The remainder of the paper is organized as follows: We first summarize the TTOSOM. Subsequently, we present the details of the design and implementation of our TTOSOM-based classifier. Thereafter, we provide the experimental results, and finally, we conclude the paper with a discussion of the results of our study.

2 The Tree-Based Topology Oriented SOM

The authors of [1] presented the Tree-based Topology Oriented SOM (TTOSOM), a technique by which the user can represent data points using prototypes, both with respect to the underlying distribution and an arbitrary tree-like topology. Since the topology can be fairly arbitrary, the TTOSOM defines a Bubble of Activity (BoA) different from the ones defined in the prior literature, both structurally and conceptually. As we can see, the map learned as a consequence of the training process is able to infer both the distribution and, simultaneously, the structured topology of the data. This was verified by extensive experiments. The strategy proposed reduces to the traditional 1-dimensional SOM when the tree is a linear sequence of nodes. In other words, the traditional SOM is a special case of the family of ANNs proposed in [1].

Acquiring information about a set of stimuli in an unsupervised manner usually demands the deduction of its structure. In general, the *topology* employed by any ANN possessing this ability has an important impact on the manner by which it will "absorb" and display the properties of the input set. Consider the following example: A user may want to devise an algorithm that is capable of learning a triangle-shaped distribution as the one depicted in Figure 1.

[1] This can be done in numerous ways, but we have chosen to do it using a simplistic Euclidean criterion.

The SOM tries to achieve this by defining an underlying grid-based topology and to fit the grid within the overall shape, as shown in Figure 1a (duplicated from [12]). However, according to the authors of [1], a grid-like topology does not naturally fit a triangular-shaped distribution, and thus, one experiences a deformation of the original lattice during the modeling phase. As opposed to this, Figure 1b, shows the result of applying the TTOSOM. As the reader can observe from Figure 1b, a 3-ary tree seems to be a far more superior choice for representing the particular shape in question.

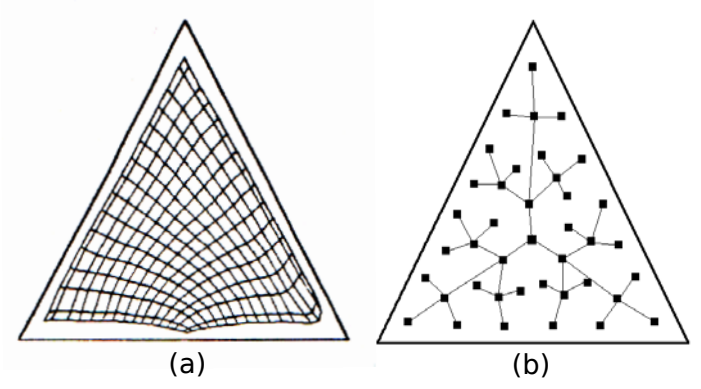

Fig. 1. How a triangle-shaped distribution is learned through unsupervised learning. (a) The grid learned by the SOM. (b) The tree learned by the TTOSOM.

On closer inspection, Figure 1b depicts how the complete tree fills in the triangle formed by the set of stimuli, and further, seems to do it *uniformly*. The final position of the nodes of the tree suggests that the underlying structure of the data distribution corresponds to the triangle. Additionally, the root of the tree is placed roughly in the center of mass of the triangle. It is also interesting to note that each of the three main branches of the tree, cover the areas directed towards a vertex of the triangle respectively, and their sub-branches fill in the surrounding space around them in a recursive manner, which the authors of [1] identify as being a holograph-like behavior. The results of [1] also showed how the TTOSOM can be used to obtain the skeleton structure of an image being examined, and its Pattern Recognition (PR) capabilities.

3 The TTOSOM-Based Classifier

Zhu, in [15], proposed the concept that clustering algorithms could be employed to perform pattern classification. As per his solution methodology, one alternative is to perform classification by applying the so-called Cluster-then-Label method. Prior research to the latter approach includes [3,4,9], among others.

Given a clustering algorithm $\mathcal{A}_C$, a set of labeled instances $\mathcal{X}_L$, a set of unlabeled instances $\mathcal{X}_U$, and a supervised learning algorithm $\mathcal{A}_S$, the Cluster-then-Label method works as follows: First, we identify the clusters of the input manifold using the clustering algorithm $\mathcal{A}_C$. Secondly, we determine which of the labeled samples fall in each cluster. For each cluster we determine a decision boundary based on the supervised algorithm $\mathcal{A}_S$, and the labeled samples assigned to that cluster, which, in turn, allows the prediction of the label of every cluster. Finally, each uncategorized item is labeled according to the predicted class of the cluster in which it is contained.

According to the author of [15], the performance of this approach is dependent on the capabilities of the clustering algorithm to mimic the properties of the original data distribution.

Our aim is to devise a classifier that works in 2 stages. First, we learn the stochastic properties of the data in an unsupervised manner. Secondly, we use some labeled items to tag the decision regions created previously. The resultant TTOSOM-based classifier is described in Algorithm 1.

In order to learn the decision boundaries, the TTOSOM algorithm is employed to train a tree structure so as to mimic the properties of the distribution of data points of *all* the classes, which is done without the necessity of providing the actual class labels of the items. This corresponds to line 1 of Algorithm 1. The output of this initial phase is a TTOSOM tree structure, where each of the neural nodes are optimally placed in the feature space so as to glean the properties of the data distribution. Our hypothesis is that these neurons represent regions of the hyper-space belonging to the same taxonomy, whose label is unknown. The problem then is to accurately guess the actual label of that taxonomy.

In the subsequent phase (see line 2 of Algorithm 1), our classifier determines which subset of the labeled instances are represented by each neuron. In an ideal scenario, where a neuron is the Best Matching Unit (BMU) of instances belonging to the *same* category, the decision of tagging the unlabeled instances falling into the region will be trivial. Unfortunately, as the authors of [3] point out, the latter does not occur necessarily. For this reason a general mechanism is required which permits the *a posterior* decision about the class to be assigned to each neuron. We thus maintain a statistical record of the number of instances belonging to each category that fall in a particular region where a neuron is the BMU.

The next phase (see line 3 of Algorithm 1), consists of a *supervised* phase in which class labels are assigned to each neuron in the tree. From a statistical perspective, when the functions that dictate the probability of finding an item in a certain region of the hyperspace are known, the problem of deciding the category of a particular sample in the area can be optimally determined by the function which maximizes its probability where the query item is positioned. However, as per our problem statement, these probability density functions are not known, and so if one employs an approach like the one described above, we must have an "approximation of sorts" of such functions. Fortunately, there is a simple way to have a rough estimation of the probability functions, i.e., by

using the information provided by the labeled training set. Each neuron in the tree is thus assigned a label based on the k-Nearest Neighbor (k-NN) rule [5]. On closer inspection, the label of each neuron will be the one which occurs more frequently among the k nearest samples, where k is the number of data points for which the particular neuron is the BMU.

Algorithm 1. `TTOSOM-Build-Classifier`($\mathcal{X}_U$,$\mathcal{X}_L$)

Input:
i) $\mathcal{X}_U$, the set of unlabeled instances.
ii) $\mathcal{X}_L$, the set of labeled instances.
Output:
i) A set of labels $\mathcal{Y}_U$ of the unlabeled samples $\mathcal{X}_U$.
Method:
1: Train a TTOSOM tree using $\mathcal{X}_U \cup \mathcal{X}_L$.
2: Determine the subset $\mathcal{X}_L^i \in \mathcal{X}_L$ for which the neuron i is the BMU.
3: Label each neuron using $\mathcal{X}_L^i$ and the k-nearest neighbors rule, where $k = |\mathcal{X}_L^i|$.
4: Label each sample in $\mathcal{X}_U$ *as per* the label of its respective BMU.
End Algorithm

The final step (line 4 of Algorithm 1) consists in predicting the class label of each of the unlabeled instances. In our method, this is done by taking a particular instance referred to as the "query" instance and finding its, BMU, i.e., the closest neuron in the feature space, which is basically the notion of a Vector Quantization (VQ) query. The class label of the query instance will be same as the class label of the neuron which is "representing" it. Given the nature of the TTOSOM, some of the neurons act as a "joint" within the tree, reflecting the concentration of other smaller clusters in its vicinity. It is likely, that these joints may not represent any sample in particular, and therefore, one needs an additional assumption in order to define its class label. In our case, we have simply decided to exclude them from the competitive learning process. In that sense, the search for the BMU in the classifier is slightly different from the one utilized by the TTOSOM (and inherited from the SOM). In this case, the label of the neuron is examined, and when it is undefined, the respective neuron is excluded from the "competition" process, which is a phenomenon that we call *supervised BMU search*.

4 Experimental Setup

The Classifiers: The classifiers considered in this study are 5 supervised classifiers, namely, Bayesian Network (BN), Naïve Bayes (NB), C4.5, k-NN and LVQ1, and 2 "semi-supervised" classifiers, namely, the TTOSOM and the SOM. The reader may consult [5] for a general overview of these schemes.

Performance Metrics for Comparing Classifiers: In this study, we shall utilize the most simple and widely-used performance metric, i.e., the accuracy of

the classifier [6]. Even though the accuracy measure is, in many contexts, inadequate, our experience is that the inferences gleaned from using it are identical to those obtained by using a more elaborate measure such as the Area Under the ROC Curve (AUC). For a comprehensive examination of metrics for quantifying the quality of a classifier, the interested reader is requested to consult [2,14].

Stochastic Sampling: In this study, we use the technique referred to as "Stratified 10-fold cross-validation". Here, the training samples are roughly divided into 10 equal partitions. Each fold is further used for testing the classifier, while the remainder 9 are employed for training. The process is then repeated for each of the folds. The term stratified, comes from the statistical concept known as "stratified sampling", which is a sampling method that draws items from the different categories so as to obtain relatively homogeneous subgroups.

The Datasets: To test the ability of the TTOSOM for classifying items belonging to the real world domain, we have 6 datasets from the UCI Machine Learning repository [8]. These datasets are Iris, Wisconsin Diagnostic Breast Cancer (WDBC), Wine, Yeast, Wine_Quality, and Glass.

The datasets used in these experiments have different numbers of output classes, ranging from 2 to 10. Additionally, their features pertain primarily to the continuous domain, whose dimensions varies from 4 up to 30. Table 1 describes the different aspects of each dataset, including its name, number of instances, number of attributes, number of output classes and problem type.

Table 1. Datasets selected for the comparison of the classifiers

Dataset	Instances	Attributes	Classes	Problem Type
Iris	150	4	3	classification
WDBC	569	30	2	classification
Wine	178	13	3	classification
Yeast	1,484	8	10	classification
Wine_Quality (red)	1,599	11	6	classification/regression
Glass	214	9	6	classification

The Parameters: The respective parameters for the algorithms were rendered to be the same across all the different datasets, and no algorithm possessed parameter values that were tuned for the datasets. In particular, the 3 strategies based on VQ, i.e., the TTOSOM, the SOM and the LVQ1 utilized the same number of iterations $(50,000)$. Additionally, they all used the same initial learning rate (0.5), and the radius of the BoA was chosen in such a way that initially, all the neurons were considered as part of the BoA, i.e., twice the depth of the tree in the case of the TTOSOM, and the width plus the height in the case of the SOM. Observe that LVQ1, as defined in [11], does not consider a BoA. As well, the three schemes utilized the same (linear) decaying schedule for its parameters.

5 Results

Comparison to Other Classifiers: The results of the performance of the different classifiers (columns) across all the dataset (rows) is summarized in Table 2. Specifically, we are interested in the performance of our classifier on problems across a diversity of domains in which labeled and unlabeled data is available[2]. For example, Table 2 shows that the TTOSOM classifier, using *only* 15 neurons, is able to accurately predict with an accuracy of 89.33% the correct label of the instances belonging the wine dataset. On the other hand, the SOM classifies correctly the same dataset with an accuracy of only 67.98%.

One possibility for quantifying the quality of our method is to consider the family of classifiers inheriting the VQ mechanism. One such strategy that belongs to the supervised family is the LVQ1, while the SOM and the TTOSOM primarily learn the distributions using the unsupervised learning paradigm. The three classifiers utilized the same parameters, which are described in Section 4. Besides, while the LVQ1 and the SOM utilized 128 neurons, the results shown for the TTOSOM include *only* 15 neurons. As per our results, the TTOSOM, using only a small percentage of the neurons used in the SOM and LVQ1 (almost 10%), outperforms their recognition capabilities in *all* six datasets.

Apart from the above, observe that the classification results offered by the TTOSOM are comparable to the ones obtained by the k-NN. However, both approaches present important differences in how they perform learning. First of all, the k-NN, being a supervised classifier, requires *all* the instances to be properly labeled. Secondly, due to its "laziness", the computations for the k-NN are left until a query is performed, which implies that the *whole* manifold is visited so as to create the ordering of the samples, as per their proximity to the query sampl6 years since the date of the TRes. On the other hand, the TTOSOM only requires a small subset of the tagged labels, and is able to learn from unlabeled samples. Also, the query is done by using the TTOSOM tree and the respective labels of the neurons, and only requires the comparison with the total number of neurons, which is usually significantly smaller than the entire dataset. Even though our method internally uses the k-NN to tag the neurons, we note that this is done only once, i.e., when the tree is being learned, and furthermore, the computations are performed only for each neuron instead of the whole dataset.

Another perspective by which we can compare the schemes is to consider the "most" competitive supervised classifiers. In this case, except for the LVQ1, they outperformed the accuracy produced by the unsupervised strategies. This is an expected behavior, because the supervised classifiers had access to the class labels of *all* the instances. However, in environments where only few tags are available, traditional supervised classifiers struggle to extract useful information from unlabeled instances. Indeed, experiments performed by Gabrys *et al.* [9], showed that when a sufficiently large number of labeled instances were utilized,

[2] Our hypothesis is that one should use as much labeled data as is available. Since the datasets mentioned above are all composed of labeled instances, we have opted to use *all* this information in the "supervised" phase of our algorithm.

the semi-supervised schemes included in their study achieved levels of accuracy that were comparable to the ones obtained by the supervised classifiers that incorporated a much higher number of labeled samples.

Table 2. General classification results of the methods investigated, reported in terms of accuracy (shown in percentages)

Dataset	TTOSOM15	BN	NB	C4.5	k-NN	LVQ1	SOM
iris	96.67	92.67	96.00	96.00	95.33	96.00	84.67
wdbc	93.32	95.08	93.15	93.15	96.66	92.09	90.51
glass	67.29	71.96	49.07	67.76	67.76	61.22	63.08
wine	89.33	98.88	97.19	93.82	94.94	74.16	67.98
yeast	54.18	56.74	57.61	55.86	54.78	24.33	46.16
wine_quality	51.91	57.72	55.03	62.91	57.79	44.15	49.59

Effect of the Number of Neurons: Another set of experiments were conducted so as to observe the effect of the *number* of neurons on the classification accuracy. To test this, we systematically increased the size of the TTOSOM tree. In order to retain the desired property that, initially, all the neurons are considered as part of the BoA, in each case we adjusted the radius to be twice the depth of the tree. Even though the size of the tree was increased, we decided to maintain the number of training iterations to be unchanged.

We identified an increase in the performance as the number of neurons is increased. For example, for the wine dataset, an accuracy of 64.61% was obtained when using 15 nodes, and increased to 76.40% when using 1023 nodes. Similarly, for the glass dataset, we obtained an accuracy of 69.16% when we used 15 nodes, which increased to 71.96% when the number of nodes was 127.

Additionally, we noted that a lesser number of neurons, which implies a lower computational requirement, outputs a fairly good approximation to the one offered by the reported supervised classifiers.

Changing the Distance Measure: In all the results presented so far, we assumed that the data was previously normalized. Specifically, the classifiers utilized the so-called Local Normalization [7], in which the range of every dimension was scaled to be between 0 and unity so as to have them equally weighted. We performed additional experiments so as to observe how the technique behave if we maintain all the parameters at their original values, and simultaneously not perform any type of normalization prior to the training process.

As a general remark we note that one observes differences with respect to the case when the data was normalized. For example, in the *glass* dataset it was possible to obtain an accuracy of 71.96% when using 127 neurons which is an index equivalent to the one provided by the best supervised classifier (BN) for this specific problem domain. It is even more interesting to see that when the number of neurons was increased to 1,023, the accuracy obtained was 74.30%, which is the *best* reported accuracy obtained for the glass dataset, when one includes *all* the supervised classifiers displayed in Table 2.

However, we have noticed as well, that in some problem sets, as in the case of the *wine* dataset, the classification accuracies are inferior to those obtained when an *a priori* normalization was invoked.

Our explanation for this phenomenon is that, when we do not normalize the feature vectors before processing them, the classifier weights those features with larger ranges for its values, more, and in certain cases it happens that these features are exactly the ones that help to advantageously discriminate between the different categories. This reasoning also explains the scenario when poorer results are obtained. This is apparently a consequence of weighting certain features (i.e., those which possess a high variance) more, i.e., those which offer inadequate discriminating aspects. Those features do not provide information that is too useful for effectively building the discrimination regions.

Using Trees Other Than Binary Trees: All the experiments presented previously in this section have employed a binary tree structure. To further investigate the power of the TTOSOM, we performed another set of experiments so as to test the effect of using trees with a higher branching factor, i.e., the number of children that a particular node had.

In particular we tested the algorithm using trees with a branching factor of 3. As far as we could observe, there were no noticeable changes in accuracy when the branching factor per node is increased from 2 to 3.

In [1], when we focused on the clustering properties on the TTOSOM, we showed how different branching factors led to a "better representation" of certain shapes. By better representation, in this case, we meant that the basic properties of some objects were preserved, so that the human eye could perceive the essential characteristics of the original object by merely looking at the learned structure. The above mentioned paper included examples, including a triangle and a rectangle, which were represented in a superior manner using specific branching factors (c.f., the representations in [12], which correspond to neural structures for the triangle using a grid and a line, respectively).

The clustering property mentioned above suggests that the symmetry presented in some data sets could be better exploited by a TTOSOM-based classifier using the adequate branching factor. However our preliminary evidence shows us that at least for the *real-world dataset* that we tested, the classifier is not noticeably affected by incrementing the number of branches in the tree. Instead, the number of neurons utilized, regardless of the branching factor of the tree, seems to be more pertinent when it concerns the resultant accuracy. This certainly is an avenue for further research.

6 Conclusions

The purpose of this paper was to design and present an experimental analysis of a novel PR scheme based on the TTOSOM. Our classifier combined the information provided by labeled and unlabeled instances simultaneously.

Our experimental results showed that the TTOSOM classifier possesses an improved classification accuracy in comparison to other VQ-based classifiers. Additionally, these accuracies are comparable to the one attained by the

state-of-the-art schemes, even when the number of neurons utilized is only a small fraction of the cardinality of the dataset.

Moreover, increasingly superior recognition capabilities could be obtained when training trees with a larger number of neurons. In particular, our results suggest a "monotonic" improvement of the mean classifier performance as the size of the tree is increased. We believe that this occurs because of the desirable properties of the TTOSOM to mimic the underlying distribution of the points, and its capability to represent the stochastic and structural characteristics more accurately by utilizing a larger tree.

References

1. Astudillo, C.A., Oommen, B.J.: Imposing tree-based topologies onto self organizing maps. Information Sciences 181(18), 3798–3815 (2011)
2. Baeza-Yates, R.A., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston (1999)
3. Dara, R., Kremer, S., Stacey, D.: Clustering unlabeled data with SOMs improves classification of labeled real-world data. In: Proc. of the 2002 International Joint Conference on Neural Networks, IJCNN 2002, vol. 3, pp. 2237–2242 (2002)
4. Demiriz, A., Bennett, K., Embrechts, M.: Semi-supervised clustering using genetic algorithms. In: Artificial Neural Networks in Engineering (ANNIE 1999), pp. 809–814 (1999)
5. Duda, R., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley-Interscience (2000)
6. Fawcett, T.: An introduction to ROC analysis. Pattern Recogn. Lett. 27(8), 861–874 (2006)
7. Fayyad, U., Grinstein, G.G., Wierse, A.: Information Visualization in Data Mining and Knowledge Discovery. Morgan Kaufmann Publishers Inc., San Francisco (2001)
8. Frank, A., Asuncion, A.: UCI machine learning repository (2010), `http://archive.ics.uci.edu/ml`
9. Gabrys, B., Petrakieva, L.: Combining labelled and unlabelled data in the design of pattern classification systems. International Journal of Approximate Reasoning 35(3), 251–273 (2004), Integration of Methods and Hybrid Systems
10. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Computing Surveys 31(3), 264–323 (1999), `citeseer.ist.psu.edu/jain99data.html`
11. Kohonen, T.: Improved versions of learning vector quantization. In: 1990 IJCNN International Joint Conference on Neural Networks, vol. 1, pp. 545–550 (June 1990)
12. Kohonen, T.: Self-Organizing Maps. Springer-Verlag New York, Inc., Secaucus (1995)
13. Lazebnik, S., Raginsky, M.: Supervised learning of quantizer codebooks by information loss minimization. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(7), 1294–1309 (2009)
14. Manning, C.D., Raghavan, P., Schütze, H.: An Introduction to Information Retrieval. Cambridge University Press, Cambridge (2009), `http://nlp.stanford.edu/IR-book/information-retrieval-book.html`
15. Zhu, X., Goldberg, A.B.: Introduction to Semi-Supervised Learning. Morgan & Claypool Publishers (2009)